Military Technical College
Kobry El-kobbah,
Cairo, Egypt

5th International Conference
on Electrical Engineering
ICEENG 2006

A PROPOSED ARCHITECTURE FOR STREAM CIPHERS USING LFSR AND FCSR

N. Shaker[*], K. Shehata[**], T. Elshafiey[***], A. Alshobaki[****]

ABSTRACT

A major class of applied stream cipher systems makes use of combining a number of Linear Feed Back Shift Registers (LFSR) to achieve reasonable cryptographic specifications. The Feed back with carry shift register (FCSR) is a new device to be used in the structure of stream ciphers in addition to or as a substitute for the LFSR. In this paper, we investigate the advantages and limitations of combining both LFSR and FCSR in one stream cipher design. A proposed novel design for a stream cipher architecture – using both LFSR and FCSR – will also be introduced and its cryptographic specifications will be evaluated.

KEY WORDS

Encryption, Stream Cipher, Linear Complexity, 2-adic Complexity, Algebraic Degree and Correlation Immunity.

[*] Comsec. Consultant.
[**] Prof. AAST, Cairo – Egypt.
[***] Assisted Prof., MSA, Cairo – Egypt.
[****] Graduate student, AAST, Alex. – Egypt.

1- INTRODUCTION

Applying cipher systems is the most proven techniques for securing communication links. Due to the current spread of communication and information technology, the society requires ultra-high speed data communications links while the requirement for achieving encryption security became permanent obligation. Stream cipher systems have many advantages over block ciphers such as simplicity of architecture, short processing delay and simplicity in security assessment, consequently, stream ciphers are more popular in proprietary applications of cipher systems.

Stream cipher architecture is correlated from long time ago with the introduction of linear feedback shift register (LFSR) which is a simple and very fast device that produces pseudorandom bits with good statistical properties and large period.

In the standard model of the stream cipher, the outputs of several independent LFSRs sequences are combined together using a nonlinear Boolean function that produces the final keystream. The main purpose of the nonlinear combining function is to make the keystream difficult for the cryptanalyst to predict.

In 1997, Klapper and Goresky [1] proposed a new type of random number generator called the feedback with Carry Shift register (FCSR). This register is based on an algebraic framework for analysis analogous to that in LFSR. This algebraic structure is based on algebra over the 2-adic numbers in which the sequences generated by FCSR that analyzed, in the same way as the algebra over finite fields can be used to analyze LFSR sequences.

This paper is divided into four parts. In the second part, we give brief descriptions of the basic building blocks of the stream cipher systems with focus to the FCSR as a newly introduced component, as well; we highlight the concatenation of LFSR and FCSR. In the third part, we describe the idea behind the proposed algorithm and we evaluate its cryptographic parameters to show the figure of merits of our design. In the fourth part we came up with the conclusion of this work.

2- BASIC BUILDING BLOCKS OF STREAM CIPHERS

2.1 Linear Feedback Shift Register (LFSR)

Linear Feedback Shift Register is a well known device that implement linear feedback function, for generating periodic binary sequences. The linear feed back function combines a set of the contents of register cells (called the feedback taps) and then injects the resultant output to the first cell of the register. The choice of the feedback taps is defined by a characteristic polynomial $Q(X)$ with a degree k here k equals the number of stages in the register. If $Q(X)$ is primitive polynomial, then the obtained sequence of the LFSR is guaranteed to have the maximal period which equals to $2^k - 1$. Such maximal sequence is called m -sequence.

The linear complexity LC of a binary periodic sequence A is the length (i.e. the number of cells) of the smallest LFSR that can generate A [2].

An unknown LFSR can be completely cryptanalyzed using just $2LC$ bits of its output. Cryptanalysis means defining the unknown tap configuration of the register as well as its unknown initial contents of register cells i.e. its unknown initial state. Consequently, LC is a very significant cryptographic specification for this class of stream cipher systems.

2.2 Feedback with Carry Shift Register (FCSR)

The architecture of the FCSR is given in Fig. 1. It consists of r register cells and additional $r - 1$ memory cells for storing the carry variables. The feedback connections are given by the coefficients q_1, q_2, …, q_r that are defined by the 2-adic integer q which has the binary expansion given by:

$$q = q_0 + q_1 2 + q_2 2^2 + \ldots + q_r 2^{r,} \qquad \text{where } q_i \in \{0, 1\} \tag{1}$$

q is called the connection integer of the FCSR [1].
If the binary contents of the register at any given time are (a_{r-1}, a_{r-2}, …, a_1, a_0) and the contents of the memory cells are (c_{r-1}, c_{r-2}, …, c_1), then the operation of the shift register is defined as follows:

1. Take an integer sum $\sigma_j = a_j + a_0 q_j + c_j$, with $1 \le j \le r - 1$
2. Put $a_{j-1} = \sigma_j \bmod 2$
3. The higher order bit is used to replace the memory cell $c_j = \lfloor \sigma_j / 2 \rfloor$.

while the number of register cells r is given as $r = \lfloor \log_2 (q+1) \rfloor$.
Suppose a sequence A, then the *2-adic complexity* $\phi_2(A)$ of the sequence A is the smallest FCSR (the number of cells in the basic shift register) whose output is the sequence A [1].
The generated binary sequence A out of the FCSR will be periodic in case that the following necessary and sufficient conditions will be satisfied [1]:

1. q is prime and 2 is primitive root modulo q.
2. Initiate the FCSR with non degenerate initial state.

Degenerate state is the state which makes the FCSR generates 0's or 1's sequences. If the memory cells c_i ($i = 1, ..., r - 1$) are provided only when the corresponding feedback tap q_i is nonzero, then there are only two degenerate states;

3. All shift register and memory cells = 0 (Zeros state).
4. All shift register and memory cells = 1 (Ones state).

Any other initial state will make the FCSR to generate eventually periodic sequences. The length of the generated periodic sequence of the FCSR is given by $T = q - 1$. This is true only if q is prime and 2 is primitive root modulo q. The reader may refer to [1] for more detailed elaboration of the theory of FCSR.
A modification to the above introduced architecture of the FCSR is shown in Fig. 2, in which an input is added to the FCSR. This slight modification is interpreted mathematically as the division of an input integer by a constant integer q imbedded in the FCSR structure [3]. For example, if the input of the above circuit is $A = (a_0, a_1, \ldots)$, then the output is given by $S = A / q$ [3].
We call such FCSR Galois architecture circuit by FCSR Function (divisor-box in [4]) with one input and one output variable. Such FCSR function is illustrated in Fig. 3.

2.3 Concatenation of LFSR and FCSR

One of the applications of the FCSR function is to be used as a nonlinear filter for the linear sequences that generated from the LFSR. This filtering process is achieved by concatenating the LFSR to the FCSR Function as shown in Fig. 4.

The resultant output sequence of the concatenated LFSR and FCSR will satisfy the following advantageous cryptographic properties:

1. **Period**
   From Fig. 4, set the LFSR size to be k and its period is T_L. Also, set the FCSR size is k and its period is T_F. Suppose that $I_L = \sum_{i=0}^{T_L - 1} L_i 2^i$, where L_i is the LFSR output i–th bit. If T_L and T_F are coprime, then the concatenation of LFSR and FCSR generates sequences with period T_S given by [4]

$$T_S = \begin{cases} T_L \cdot T_F & if \quad \gcd(I_L, q) = 1 \\ T_L & if \quad \gcd(I_L, q) \neq 1 \end{cases} \tag{2}$$

   Moreover, there are some initial states makes $T_S = T_L$, but the number of these initial states decreased with respect to the number of initial states – which makes $T_S = T_L \cdot T_F$ – as long the LFSR and FCSR sizes are increased [4, 5].

2. **Algebraic degree**
   The algebraic degree of the output sequence S obtained by representing the output S as function of the secret key values. In that case, we assume that the only secret value is the initial vector of the LFSR ($p_0, ..., p_{k-1}$) and all the other parameters are known.
   For each i from 0 to T_s (period of S), there exists a Boolean function F_i such that $s_i = F_i$ ($p_0, ..., p_{k-1}$). Defining the sequence S is equivalent to solving the system of equations

$$F_i\,(p_0, ..., p_{k-1}) = s_i \text{ for } i = 0, 1, ..., T_s - 1 \tag{3}$$

   Under this assumption, our problem can be modeled by a system of equations of expected degree at least k, and $k + 1$ unknowns, see [5] for proof.

3. **Linear complexity**
   Following [4], the linear complexity is about 2^{2k-2}.

4. **2-adic complexity**
   The expected 2-adic complexity of S is $\phi_2(S) = 2^k + k - \frac{2^k}{k}$ [4].

2.4 A Model Structure of Stream Cipher System

Fig. 5 shows a model structure for the stream cipher systems which employs a combiner function f with multiple input driving sequences and single output sequence. The combiner function was proposed to have a memory to insure both nonlinearity and correlation immunity. The correlation immunity of this model structure reaches the maximum order (n – 1), where n is the number of input driving sequences [6].

$$Z_j = \sum_{i=1}^{n} S_{ji} + X_j \tag{4}$$

$$X_{j+1} = f_s(S_{j1}, ..., S_{jn}, X_j) \tag{5}$$

2.5 Related Definitions of Boolean Functions

A Boolean function [7] maps one or more binary input variables to one binary output variable $f : GF(2^n) \to GF(2)$ and can be represented by binary form, $f(x) \in \{0,1\}$. Sometimes it is desirable to consider a Boolean function over the set {1, –1} rather than {0, 1}. The *polarity form* of a Boolean function is denoted $\hat{f}(x)$ where $\hat{f}(x) \in \{1,-1\}$ and $\hat{f}(x) = (-1)^{f(x)}$.

If the number of zeros in the function *f*(*x*) is equivalent to the number of ones then the function is said to be *balanced*. The following terms are frequently used in this paper:

a) Linear Boolean function

A linear Boolean function L_ω (*x*), selected by $\omega \in GF$ (2^n), is given by $L_\omega(x) = \omega_1 x_1 \oplus ... \oplus \omega_n x_n$ where $\omega_i x_i$ denotes the bitwise AND of the *i*-th bits of ω and *x*, and ⊕ denotes bitwise XOR.

b) Affine function

The set of affine functions $A_{\omega,c}$ (*x*) is the set of linear functions and their complements; it is given by $A_{\omega,c}(x) = L_\omega(x) \oplus c$, where $c \in \{0, 1\}$.

c) Walsh Hadamard transform

For a Boolean function *f* the Walsh Hadamard Transform W_f is defined by $W_f(\omega) = \sum_{x \in GF(2^n)} \widehat{f}(x) \widehat{L}_\omega(x)$ Thus, each Walsh Hadamard Transform $W_f(\omega)$ is the vector dot product of the polar forms of *f* and the linear function L_ω.

d) Nonlinearity

The nonlinearity N_f of a Boolean function *f* is its minimum distance to any affine function. It is given by $N_f = \frac{1}{2}(2^n - \max(W_f(\overline{\omega})))$.

e) Correlation immunity

A function *f* is correlation immune *CI* of order *m* if and only if $W_f(\overline{\omega}) = 0$; $1 \le wt(\overline{\omega}) \le m$, where $\overline{\omega} = (\omega_n, ..., \omega_1)$, and *wt*($\omega$) is the hamming weight of ω.

Siegenthaler [8] has shown that for functions with *n* inputs and with correlation immunity of order *m* and algebraic degree *d*, it must follow that $m + d \le n$, and $m + d \le n - 1$ for balanced functions. Hence, high correlation immunity implies low algebraic complexity, consequently, low nonlinearity order. This relation between correlation immunity and algebraic degree can be eliminated by adding memory to the function as described in section 2.3.

3- THE PROPOSED STRUCTURE FOR STREAM CIPHER

3.1 Design Description

We propose here the mixing the concept of the concatenated LFSR and FCSR with the concept of the model structure introduced above. Our proposed structure is shown in Fig. 6.

3.2 Cryptographic Specifications of the Proposed Design

Our proposed design will have significant cryptographic specifications as follows:

1. **Period**

From Fig. 6, the number of driving sequences is n. The sequence periods of LFSRs and FCSRs are (T_{L1}, T_{L2}, ..., T_{Ln}) and (T_{F1}, T_{F2}, ..., T_{Fn}) respectively. Set T_{Si} is the period of the driving sequence S_i for i = 1, ..., n. Hence, the total period T_z of the generated sequence Z is the least common multiple of LFSRs and FCSRs periods [5]

$$T_z = lcm(T_{S1}, T_{S2}, ..., T_{Sn}) \tag{6}$$

Back from section 2.3, the period of the concatenation of LFSR and FCSR is one of two values. Therefore, the T_z lower bound is $T_{min} \leq T_z$, whereas T_{min} is the period when $T_{Si} = T_{Li}$ for all i = 1, ..., n. For practical stream cipher, T_{min} should be sufficiently large enough.

2. **Correlation immunity and probability**

The proposed algorithm uses a combiner with memory (as in Fig. 5), whereas the output (Z_j) and next-state (X_j) functions of the proposed design (Fig. 6) are the same as equations (4) and (5) respectively. Hence, the correlation immunity of the output function Z has maximum order (n – 1) [6].

The summation generator with two LFSR and one-bit of memory is proved to achieve maximum correlation immunity [6], in spite of that it is not entirely secure, whereas Meier and Staffelbach shows that existing correlation between the generator's output sequences and carry sequences (carry-output correlation probability is $\frac{1}{4}$) can easily be estimated by an outside party [9]. Therefore, to avoid correlation attacks, the existing of correlation between any input and output point should be eliminated. In the proposed design we can avoid this weakness by choosing a proper next-state function f_x characterized by the following properties:

- Balanced correlation probability of all inputs–outputs points;
  P[inputs-output(Z)] = P[inputs-output(X_j)] = P[output(X_j)-output(Z)] = $\frac{1}{2}$
- Balanced.
- Very high linear complexity.

3. **Algebraic degree**

The algebraic degree of the sequence Z depends on two factors; the function f_x and the algebraic degrees of its input driving sequences S_i as shown in Fig. 6.

From section 2.3, the expected algebraic degree of driving sequence S_i is at least k_i. On the other hand, the algebraic degree of the function f_x helps in determine the algebraic degree of the sequence Z. Suppose a function f_x with $n+1$ input variables, its input driving sequences are S_1, S_2, ..., S_n. Set the algebraic degree of the function f_x to be 3 (for example). Suppose the term S_2 S_3 S_n is the largest term of the function f_x (the terms of any function can be obtain from the algebraic normal form ANF representation of the corresponding function). Hence, the expected algebraic degree of the output sequence Z is at least $\geq k_2\ k_3\ k_n$.

4. **Linear complexity**

The driving sequences of the proposed design are obtained from concatenation of LFSR and FCSR whereas the corresponding LC of each sequence is near to its period

length [4]. Furthermore, the memory combiner which combine these driving sequence is a nonlinear combiner which in turn increases the *LC* of the produced sequence.
For small size of LFSR and FCSR, simulation examples of period T_z and there linear complexity *LC* for the algorithm in Fig. 6 with different number of driving sequences are shown in Table 1. Using the Berlekamp-Massey algorithm [2], it is shown that the linear complexity of output sequences is close to their periods.

5. 2-adic complexity

The 2-adic complexity ϕ_2 of a periodic sequence is given as follows. Suppose a periodic sequence $A = \{a_0, a_1, ..., a_{T-1}\}$ with period T. Let $S_T = \sum_{i=0}^{T-1} a_i 2^i$ and $N = 2^T - 1$. Set $d = \gcd(S_T, N)$, let $R = N / d$. Hence, the 2-adic complexity of the sequence A is $\phi_2(A) = \lfloor \log_2(R) \rfloor$.
Table 2, shows a simulation examples of periods and 2-adic complexity of the proposed algorithm for different input driving sequences. Every driving sequence S_i obtained from the concatenation of Q_i and q_i (the integer representation of the corresponding LFSR$_i$ primitive polynomial and the connection integer of FCSR$_i$ respectively). Each example in the table runs through all the good-keys to obtain the behavior of the 2-adic complexity. Therefore, Table 2 illustrates that the 2-adic complexity of the proposed algorithm is close to period length.

4- CONCLUSION

We have proposed here a novel architecture for stream cipher system through mixing LFSR and the nearly devised component known as FCSR.
The proposed architecture of the stream cipher system depends on combining a set of driving sequences S_i using nonlinear one–bit memory function *f*. The set of driving sequences S_i is proposed to be produced by concatenating LFSR$_i$ and FCSR$_i$.
The simulation results of the proposed architecture showed that the linear and 2-adic complexities of the generated sequence are close to the generated sequence period. The algebraic degree of the generated sequence depends on algebraic degree of the function f_x and the sizes k_i of the LFSR$_i$, where the algebraic degree of the driving sequence S_i is k_i.

5- REFERENCES

[1] A. Klapper and M. Goresky, Feedback Shift Registers, 2-Adic Span, Combiners with Memory, Journal of Cryptology, (1997).

[2] A. Menezes, P. van Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC press, (1996).

[3] A. Klapper and M. Goresky, Fibonacci and Galois Representations of Feedback with Carry Shift Registers, October 25, (2000).

[4] Francois Arnault, T P Berger, and Abdelkader Necer**,** A New Class of Stream Ciphers Combining LFSR and FCSR Architectures, INDOCRYPT 2002, LNCS 2551, pp. 22-33, Springer-Verlag Berlin Heidelberg, (2002).

[5] Alaa M. Alshobaki, Design and Implementation of Stream Cipher Based on LFSR and FCSR, Arab Academy for Science and Technology, MSc. Thesis, (2006).

[6] R. A. Rueppel, Analysis and Design of Stream Ciphers, Springer-Verlag Berlin, Heidelberg (1986).

[7] John Andrew Clark, Metaheuristic Search as a Cryptological Tool, Univ. of York, Department of Computer Science, (2001).

[8] T. Siegenthaler, Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications, IEEE Trans. IT-30 (5) 776-780, Sep (1983).

[9] W. Meier, O. Staffelbach, Correlation properties of combiners with Memory in Stream Ciphers, Journal of Cryptology, 67-86, (1992).

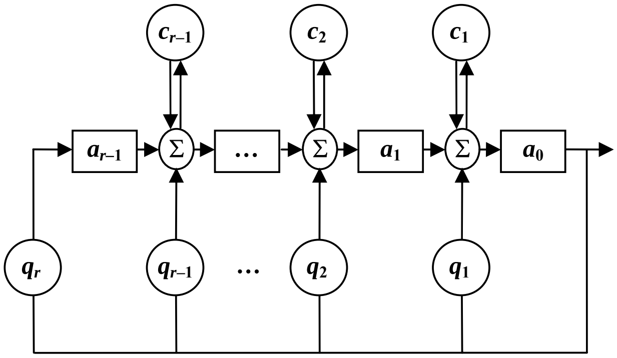

Fig.1. FCSR Galois architecture

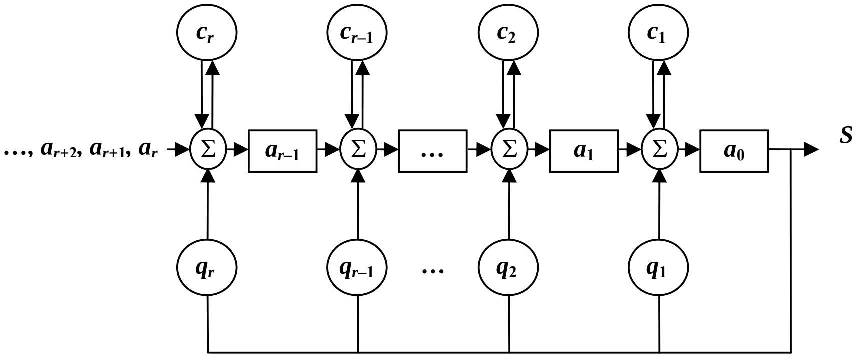

Fig. 2. FCSR Galois Function with one input output variable

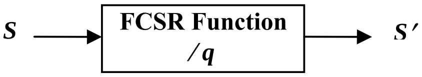

Fig.3. The FCSR Function

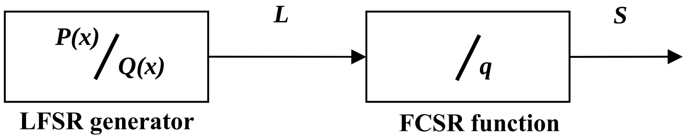

Fig.4. Concatenation of LFSR and FCSR

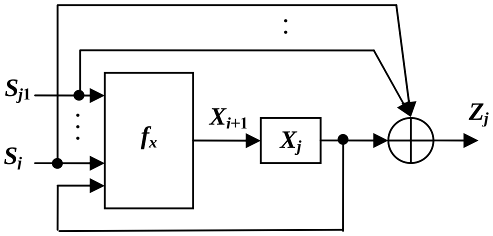

Fig.5. Combiner with Memory

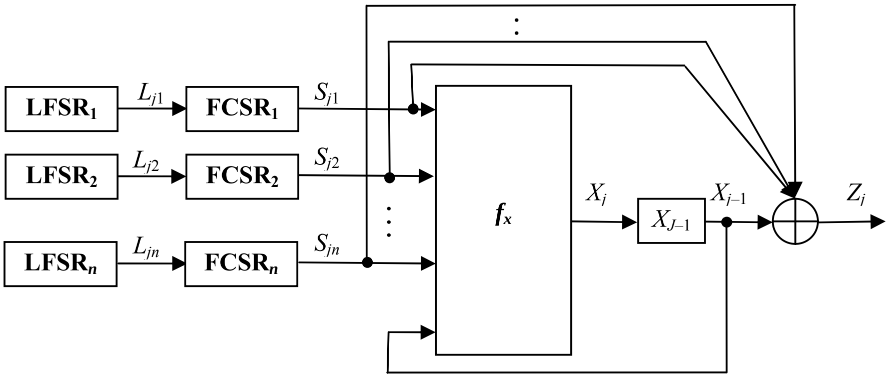

Fig.6. The Proposed Structure for Stream cipher

Table 1. Simulation examples of the period and linear complexity

Q_1	q_1	Q_2	q_2	Q_3	q_3	Q_4	q_4	Q_5	q_5	T_z	LC
11	11	19	19	–	–	–	–	–	–	630	626
19	29	55	61	–	–	–	–	–	–	13020	13016
11	11	55	59	–	–	–	–	–	–	62930	62929
11	11	19	19	25	59	–	–	–	–	18270	18268
13	13	19	29	25	37	91	101	109	181	6300	6300
67	67	19	19	97	101	13	13	25	29	69300	69291
109	421	97	67	25	61	19	19	103	163	124740	124722

Table 2. Simulation examples of the period and 2-adic complexity

Q_1	q_1	Q_2	q_2	Q_3	q_3	T_z	$\le \phi_2(z) \le$
7	5	11	13	–	–	84	78–84
11	11	19	59	–	–	6090	6084–6090
7	11	25	107	–	–	1590	1581–1590
19	19	55	61	–	–	5580	5562–5580
7	5	11	11	19	19	1260	1243–1260
11	11	19	37	67	101	6300	6259–6300